

# TP1 : Base de données

## MySQL avec l'interface phpMyAdmin

(Création, insertion, consultation et modification des données)

### Objectif

On va ici s'intéresser à la création d'une base de données en utilisant un SGBD (MySQL). Cela nous permettra en particulier d'expérimenter la création d'une base de données relationnelle en utilisant l'interface fournie avec MySQL (phpMyAdmin).

### Prise en main

Installer la dernière version disponible de EasyPHP (à ce jour Devserver 17.0)

Pour cela, accédez à la page web : <http://www.easyphp.org/>

Une fois l'installation terminée, lancez les serveurs et accédez au web local : <http://127.0.0.1:1111/>

Créer une base de donnée avec le nom : bdd

N'oubliez pas que l'interface phpMyAdmin utilise le terme de table pour désigner une relation, le terme de champ pour désigner un attribut, le terme d'enregistrement pour désigner un tuple et le terme de relation pour désigner une liaison entre tables.

### Partie 1 : Création des relations (tables)

Se positionner sur votre base de données et utiliser l'interface de phpMyAdmin pour créer vos tables. Créer les relations Etudiant, Matiere, EtudiantMat, Adresse, Enseignant de manière à ce qu'elles contiennent les attributs suivants :

**Etudiant** : numero (clé primaire), nom, prenom.

**Matiere** : code, libelle, nombre heures (cours, TP, TD confondus). Le code est la clé primaire de la relation. Attention au type de données du code de la matiere (exemple : "SL2IBD").

**Adresse** : numero de rue, nom de rue, ville, code postal, caractere ('b' pour bis, 't' pour ter, et ' ' si aucun des deux). Enfin, on pourra pour simplifier donner un "numéro généré automatiquement" à chaque adresse qui servira de clé primaire.

**Enseignant** : numero (clé primaire), nom, prenom, age, ville, nombre heures enseignees.

**EtudiantMat** : note de TP obtenue, note examen obtenue.

Attention : Créer les tables en leur donnant le type InnoDB.

Choisir les types de données qui vous semblent les plus appropriés (Texte ou Entier ...)

Spécifier si la saisie de donnée est obligatoire ou non pour un attribut (NULL/NOT NULL)

### Partie 2 : Définition des clés étrangères

On veut maintenant compléter la description de ces relations en introduisant les informations qui nécessitent de lier ces relations entre-elles (définir les clés étrangères). Pour cela il faudra rajouter des attributs (avec index) ce qui correspond à rajouter des contraintes dans la « Vue relationnelle ».

Attention : Choisir l'option RESTRICT pour ON DELETE et ON UPDATE.

1. **Etudiant**. Il faut lier la relation **Etudiant** avec la relation **Adresse** afin d'exprimer le fait que tout étudiant a une adresse. Eventuellement, un étudiant peut ne pas préciser son adresse toutefois les attributs numero, nom et prenom sont obligatoires et ne peuvent pas être NULL.  
Attention : L'attribut adresse (clé étrangère) de la relation Etudiant peut être NULL.  
 L'attribut adresse ne doit pas être unique. Expliquez pourquoi ?
2. **Matiere**. A propos de la liaison de la relation **Matiere** avec la relation **Enseignant**, afin d'exprimer le fait que chaque matiere a un unique enseignant responsable (un enseignant est responsable au plus d'une matière). Utiliser un index Unique pour la clé étrangère
3. **EtudiantMat**. La relation **EtudiantMat** sert de lien entre la relation **Matiere** et la relation **Etudiant** afin de pouvoir faire le lien entre les notes de l'étudiant et la matiere concernée.  
Attention : La relation EtudiantMat ne dispose toujours pas de clé primaire.  
 Pour définir la clé primaire, nous ne choisirons pas ici de numéroter les tuples comme on a pu le faire plus haut. Le choix de la clé servira à répondre à la contrainte suivante :  
 A propos de la liaison EtudiantMat de la relation Matiere avec la relation Etudiant, afin d'exprimer le fait que les notes sont obtenues une seule fois par un étudiant. Quelle est la clé primaire à choisir pour répondre à cette contrainte ?

### Partie 3 : Insertion de données

Pour chaque table vous cliquerez sur le bouton ou l'onglet « insérer » selon que vous soyez au niveau de la base de données ou à celui de la table concernée. L'objectif est de créer les tuples.

A priori il faut opérer dans un ordre logique : **Adresse, Etudiant, Enseignant, Matiere, EtudiantMat**.

Saisir les tuples suivants :

Adresse(1, 34, 'b', "Biranzarane", "20000", "Casablanca")  
 Adresse(2, 19, ' ', "Elfath", "10000", "Rabat")  
 Adresse(3, 11, ' ', "Biranzarane", "20000", "Casablanca")  
 Etudiant(1001, "Rziza", "Sara", 1)  
 Etudiant(1002, "Belkacemi", "Mohamed", 2)  
 Etudiant(1003, "Rziza", "Morad", 3)  
 Enseignant(1, "Azeroual", "Driss", 55, 135, "Casablanca")  
 Enseignant(2, "Boukili", "Driss", 47, 230, "Casablanca")  
 Enseignant(3, "El Ghorfi", "Rabii", 28, 100, "Rabat")  
 Enseignant(4, "Menez", "Philippe", 39, 30, "Bordeaux")  
 Matiere("SL2IBD", "Base de données", 24, 3)  
 Matiere("SL2IPI", "Programmation impérative", 36, 1)  
 Matiere("SL2IAL", "Management", 32, 2)  
 Matiere("SL2IPW", "Français", 26, 4)  
 EtudiantMat(1001, "SL2IBD", 10, 11)  
 EtudiantMat(1001, "SL2IPI", 8, 10)  
 EtudiantMat(1002, "SL2IBD", 10, 11)  
 EtudiantMat(1002, "SL2IPI", 8, 10)  
 EtudiantMat(1003, "SL2IBD", 10, 11)  
 EtudiantMat(1003, "SL2IPI", 8, 10);  
 EtudiantMat(1003, "SL2IAL", 12, 13);

## Partie 4 : Consultation et sélection de données

Utiliser le bouton « Parcourir » pour voir l'ensemble des informations contenues dans chacune des tables. Utiliser ensuite l'onglet « Rechercher » pour afficher les informations suivantes (dans les deux cas utiliser les différentes possibilités offertes):

1. Numéro et nom de rue et ville, pour toutes les adresses
2. Nom et prénom de tous les étudiants
3. Le libellé de la matière et le numéro de l'enseignant responsable pour toutes les matières
4. Code postal et ville, pour toutes les adresses (avec ou sans suppression des doublons)

### Premières sélections :

1. Affichage de tous les étudiants dont le nom est « Rziza » (toutes les informations)
2. Affichage de tous les numéros d'étudiants qui suivent la matière « SL2IBD »
3. Afficher tous les enseignants dont le prénom est « Driss ».
4. Afficher tous les noms de rues de la ville « Casablanca ».

### Sélections plus complexes:

C'est le moment d'utiliser de manière plus fine les possibilités de MySQL pour réaliser des requêtes impliquant plusieurs tables. Pour cela, se placer au niveau de la base de données et cliquer sur l'onglet « Requête »

1. Afficher tous les noms et prénoms d'étudiants qui suivent la matière « SL2IBD » ou « SL2IPI »
2. Afficher tous les noms et prénoms d'étudiants ainsi que le libellé des matières toujours pour ceux qui suivent la matière « SL2IBD » ou « SL2IPI »
3. Afficher tous les étudiants dont le nom est « Rziza », en affichant que le prénom de l'étudiant et le nom de la rue où il habite.
4. Afficher le nom du responsable de chacune des matières « SL2IBD » et « SL2IPI »
5. Afficher le nom de l'étudiant, le libellé de la matière et la note de TP de chacun des étudiants.
6. Afficher le nom de l'étudiant, le libellé de la matière et la note de TP de chacun des étudiants dont le prénom commence par « Mo »

## Partie 5 : Modification de données et de structures

Afficher le contenu d'une table en utilisant le bouton « afficher » présent pour chaque table, puis pour chacun des champs utiliser le bouton « modifier » afin de réaliser les opérations suivantes (tuple par tuple) Remarque : On peut automatiser ces requêtes avec le langage SQL (dans un prochain TP)

### Modification des données :

1. Augmenter toutes les notes d'examen de 1 point pour tous les étudiants dans toutes les matières.
2. Pour les étudiants qui suivent la matière « SL2IBD », augmenter la note de TP de 1 point.

### Modification de la structure :

Utiliser le bouton « structure » présent pour chaque table afin de vous mettre en position de travailler sur la structure d'une table. Ensuite, utilisez les boutons « modifier » ou « supprimer » accessible pour chaque champ d'une table et le bandeau d'ajout de colonne placé sous la table pour réaliser les opérations suivantes :

1. Changer le type de l'attribut « ville » de la table *Adresse* de manière à en augmenter la taille de 5 caractères.
2. Ajouter une colonne « age » à la table *Etudiant* et compléter son contenu.
3. Supprimer la colonne « age » de la table *Etudiant*